Artificial Neural Network Tuning by Improved Sine Cosine Algorithm for HealthCare 4.0

Masa Gajevic[1], Nemanja Milutinovic[1], Jelena Krstovic[1], Luka Jovanovic[1], Miodrag Zivkovic[1](✉), Marina Marjanovic[1], and Catalin Stoean[2]

[1] Singidunum University, Danijelova 32, 11000 Belgrade, Serbia
{marija.gajevic.16,nemanja.milutinovic.16,jelena.krstovic.16, luka.jovanovic.191}@singimail.rs, {mzivkovic,mmarjanovic}@singidunum.ac.rs

[2] Faculty of Sciences, University of Craiova, Str. A.I. Cuza, Nr. 13, 200585 Craiova, Romania
catalin.stoean@inf.ucv.ro

Abstract. This paper explores classification of datasets for Healthcare 4.0 using artificial neural networks which are tuned by improved sine cosine algorithm (SCA). Healthcare 4.0 themes include internet of things (IoT), industrial IoT (IIoT), cognitive computing, artificial intelligence, cloud computing, fog computing, edge computing, and other industry 4.0 procedures. Health issues identification are critical since prompt treatment improves the quality of life for individuals affected. One of the most difficult challenges for artificial intelligence (AI) is selecting control parameters that are appropriate for the situation at hand. This paper presents a metaheuristics-based method for training the artificial neural network, by utilizing the SCA.

Keywords: Healthcare 4.0 · Sine Cosine Algorithm · Artificial intelligence · Optimization · Metaheuristics

1 Introduction

Personal and public health are both necessary for living a productive life, for both the individual and society as a whole. In the last century, much effort has been made into improving health, reducing sickness, and therefore dramatically increasing life expectancy. Health is more than just the absence of sickness; it is a full condition of social, mental, and physical well-being.

Our health is crucial, as we noticed during the COVID 19 epidemic. The most important aspects are accurate and prompt diagnosis and prevention. New technologies are assisting us in making illness prevention and early diagnosis as successful as possible. Healthcare 4.0 was formed as a result of the fourth industrial revolution, which brought us the widespread application of contemporary computer resources and artificial intelligence in different industries, including healthcare [36]. In healthcare 4.0, the patient's records are stored in electronic

N. Bacanin and H. Shaker (Eds.): ICIITB 2022, ACSR 104, pp. 289–305, 2023.
https://doi.org/10.2991/978-94-6463-110-4_21

health record (EHR) repository which may be located either at a centralized or distributed locations to help the Doctors to easily access the patient's healthcare data from anywhere at any time. As this data is accessed from the database repository using an open channel, i.e., the Internet, so, security and privacy are major concerns while accessing it from any location. A large number of wearable devices (WDs) such as smartwatch, health tracker and smart glasses are used by the patients to measure the healthcare related parameters such as heart rate, blood pressure, electrocardiograph, and breath analyzer. So, the patients need to synchronize their mobile device (MD) with the WDs to monitor their health in real-time [27].

Machine learning (ML) is a sort of artificial intelligence (AI) that can anticipate outcomes reliably without explicit programming. Classification in ML is a technique of classifying a collection of input data into classes. Many ML approaches excel at classification via supervised learning. Algorithms may learn from given data and make more accurate conclusions. Recent breakthroughs in the field of machine learning (ML) have shown to be beneficial in modern medicine, solving complicated issues. However, despite many advantages, ML methods are not without their shortcomings. Overfitting, inadequate datasets, and the hard work of selecting control parameters appropriate for the task at hand are all challenges that need experience and competence on the side of the researcher handling a particular subject. Deep learning is a type of machine learning that involves artificial intelligence and so mimics how people learn. This method is critical for statistics and predictive analytics. Artificial neural networks (ANNs), which represent a model based on the human nervous system, are at the centre of deep learning. It is a collection of algorithms that work on the idea of linked neurons to transform a set of input data into a set of output data with the help of gained knowledge [6]. In this study, we look at the use of a neural network with one hidden layer.

There are two problems with neural networks:

– The neural network must be adapted for each practical problem and this boils down to finding the optimal (suboptimal) number of neurons in the hidden layer
– Initial weights and biases randomly initialized, which is often not good enough and this second problem is called ANN training

Both of these problems are NP-hard problems and very time-consuming to solve, so non-traditional solutions such as the randomly determined estimated approach are used. In this paper, to solve these two problems, we use SCA that does the following:

– Finds the number of neurons in the hidden layer
– Determines initial weights and biases

The remainder of this paper is structured according to the following. Section 2 presents an overview of preceding research, covering multilayer perceptron (MLP) in detail. In Sect. 3 the original SCA is described. The proposed improvements to the original approach and the description of the novel proposed algorithm are covered in Sect. 3.2. The experiments and a brief discussion of the attained as well as a comparative analysis with other contemporary metaheuristics are shown in Sect. 4. Finally, Sect. 5 gives a conclusion to the work and presents plans for future research in the field.

2 Background and Related Work

2.1 Neural Networks

Neural network training is very important process and plays a crucial role in building a model which will perform better. During the weight learning process, the loss function needs to be optimized. Over-fitting is a common problem in neural network training that occurs when there is a large difference between training and test accuracy.

This issue indicates that the network has learnt extremely particular data and is unable to anticipate when new data is fed into the inputs. Various regularization algorithms, including as $L1$ and $L2$ regularization [39], dropout [47], drop connect [51], batch normalization [30], early stopping, and data augmentation, can be used to eliminate the issue of over-fitting.

Artificial neural networks may be used to solve complex issues in a variety of disciplines. When dealing with supervised or unsupervised machine learning problems, ANNs can provide respectable outcomes. As mentioned in [26], these challenges include machine perception difficulties in which it is not possible to individually understand the collection of accessible main characteristics. As a result, ANNs have been widely used in the implementation of pattern recognition, classification, clustering, and prediction issues. In the medical area, for example, many types of ANNs were used for diagnostics [26] and categorization of cardiac disorders or diabetes. The use of ANNs shortened the time necessary for diagnostics by processing enormous amounts of data during the ANN training.

A single-layer perceptron is the most abstract kind of ANN (SLP). According to [29], SLPs have only two layers: one input layer and one output layer. Unfortunately, as explained in [40] this sort of ANN is incapable of processing nonlinearly separable patterns efficiently. Later, multilayer perceptrons were presented as a model with several layers (MLP). By using one or more hidden layers, this type of neural network solves the shortcomings of the SLP model. MLPs are likely the most common type of ANNs today, with benefits such as learning capacity, parallel processing, and robustness. The ability to generalize is the most significant aspect of MLPs. MLP with a single hidden layer (SHL) networks are analyzed in this research with the purpose of optimizing the number of hidden units in the hidden layer as well as the connection weights and biases. The capabilities of any ANN can be significantly boosted depending on the learning approach used for network training. There are two basic approaches to supervised training techniques: gradient-based and stochastic methods [40]. Back-propagation is the most often used gradient-descent method nowadays. Because of the exploitation tendency, it may be used as an algorithm for local search. However, because the aim is to identify the global optimum, the optimizer used should strike a balance between exploration and exploitation. The exploration phase is responsible for searching through the undiscovered portions of the search space, whereas the exploitation phase is responsible for focusing on the areas that have previously been examined. MLP networks are a kind of feedforward neural network (FFNN). The FFNNs are made up of a group of neurons that serve as processing components. These neurons are organized into completely linked layers. MLP is made up of three types of parallel layers: input, hidden, and output. MLP neurons are configured in a one-directional mode. The links with the set weights connect the layers. Every neuron is capable of performing two basic functions: summation and activation. Equation 1 gives the summation function, which is made up of the products of the input values, allocated weights, and bias:

$$S_j = \sum_{i=1}^{n} \omega_{ij} I_i + \beta_j \tag{1}$$

where n stands for the number of input values, I_i represents the input value i, ω_{ij} is the connection weight, and finally, j denotes the bias term.

The activation function is applied on the Eq output 1. There are numerous types of activation functions that may be used, for example, an S-shaped curved sigmoid function, which is given by Eq. 2:

$$f_j(x) = \frac{1}{1 + e^{-S_j}} \tag{2}$$

The network's performance is assessed using the loss function. The MSE, or mean-squared error loss, is a popular loss function that calculates the sum of the squared distances between the actual and predicted classes as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3}$$

For example, if the data contains two features, corresponding to three neurons in the input unit, and the hidden layer contains three hidden units, the neural network may be represented as:

$$S_1 = \begin{bmatrix} \omega_{11} & \omega_{21} \\ \omega_{12} & \omega_{22} \\ \omega_{13} & \omega_{23} \end{bmatrix} \times \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} + \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} \tag{4}$$

2.2 Metaheuristics Optimization

Recent flourishing of metaheuristics algorithms, especially swarm intelligence techniques, is related to their success and efficiency in solving NP-hard problems. The existence of a large number of metaheuristics algorithms is justified by the no free lunch theorem [52], that states that there is no algorithm that is capable to obtain the best results on every possible optimization problem. Therefore, the scientists have experimented with wide range of optimization algorithms on a variety of practical problems. Some of the most promising domains include medical diagnostics support [14,20,24,33,43], wireless sensor network tuning [4, 9,12,48,57,66], stock price estimations [16], as well as intrusion detection and security domain [1,31,41,45,55,56,60,65] and plant classifying task [17].

Other successful applications of metaheuristics optimizers include tuning of the cloud, edge and fog computing [2,5,15,23,46,59], feature selection challenge [8,19,22,32,37,49,61], dropout regularization [11], a variety of COVID-19 applications [25,58,62–64], tuning artificial neural networks [3,6,7,10,13,18,44], text clustering [21,50] and cryptocurrency price forecast [42].

3 Sine Cosine Algorithm

3.1 Basic SCA

In general, population-based optimization algorithms begin the optimization process with a collection of random solutions. An objective function evaluates this random set periodically, and it is improved by a set of rules that serves as the kernel of an optimization strategy. Because population-based optimization approaches hunt for optimization issue optima stochastically, there is no assurance that a solution will be found in a single run. However, the likelihood of obtaining the global optimum grows as the number of random solutions and optimization steps (iterations) increases.

Regardless of the variations across algorithms in the field of stochastic population-based optimization, the optimization process is divided into two phases: exploration and exploitation. An optimization method in the first phase suddenly merges the random solutions in the set of solutions with a high rate of unpredictability to locate the promising areas of the search space. However, there are progressive modifications in the random solutions throughout the exploitation phase, and random variations are far lower than during the exploration phase.

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times |r_3 P_i^t - X_i^t| & r_4 < 0.5 \\ X_i^t + r_1 \times \cos(r_2) \times |r_3 P_i^t - X_{i_i}^t| & r_4 \geq 0.5 \end{cases} \tag{5}$$

r_4 is a random number in [0, 1]
where X_i^{t+1} represents the setting of the current option in the i-th measurement at the t-th model, $r_1/r_2/r_3$ are arbitrary values, P_i represents the placement of the location factor in the i-th dimension, and || represents the absolute worth.

SCA has four primary parameters, as shown by the equations above: r_1, r_2, r_3 and $r4$. The argument r_1 specifies the next location regions (or movement direction), which may be inside or outside the space between the solution and the goal. The r_2 option specifies how far the movement should be directed towards or away from the target. The parameterr_3 assigns random weights to the destination in order to stochastically accentuate ($r_3 > 1$) or deemphasize ($r_3 < 1$) the influence of desalination in determining distance. Finally, in Eq. (5), the parameter r_4 alternates between the sine and cosine components. As a result of using sine and also cosine in this solution, this algorithm is called sine cosine algorithm (SCA).

The cyclic arrangement of the sine and cosine functions allows one solution to be repositioned around another. This ensures that the space defined between two solutions is exploited. The solutions should be able to seek beyond the space between their respective destinations in order to explore the search space. This is accomplished by varying the range of the sine and cosine functions.

After all, the pseudo code of the SCA algorithm is presented in the complying with figure:

Algorithm 1. SCA Pseudocode

Initialize a set of search agents (solution)(X)
Do
 Evaluate each of the search agents by the objective function
 Update the best solution obtained so far (P=X*)
 Update r_1, r_2, r_3, and r_4
 Update the position of search agents using Eq. (5)
While (t< maximum number of iterations)
return the best solution obtained so far as the global optimum

The pseudo code illustrates how the SCA algorithm begins the optimization process with a collection of random solutions. The algorithm then keeps the best solutions found thus far, assigns them as the target point, and updates additional solutions in relation to them. Meanwhile, as the iteration counter grows, the ranges of sine and cosine functions are adjusted to highlight exploitation of the search space. By default, the SCA algorithm ends the optimization process when the iteration counter exceeds the maximum number of iterations. However, any other termination condition, such as the maximum number of function evaluations or the correctness of the determined global optimum, might be considered [38].

3.2 Improved SCA

A simple modification has been added to the original implementation of the SCA, to enhance the capabilities of the basic version of the algorithm. Each solution has been associated with a *trial* variable, that is initially set to 0, and incremented after each iteration. In case the particular individual didn't improve after k iterations, it will be removed from the population and replaced by a new, random generated individual. Due to the small number of individuals and iterations, k was fixed to 2 in the following simulations. Modified metaheuristics has been simply named improved SCA-ISCA.

4 Experiments and Comparative Analysis

The following Section aims to describe the conditions in which the test cases where performed and the performance was validated. The proposed solution is compared to other high-performing solutions from the field resulting in a tabular summary of all results that is used for comparison. The choice of datasets was specific to due the existence of reference values provided in the [35]. Note that the referenced work applies the testing over a larger group of datasets, while the presented research focuses on the following: liver disorder https://archive.ics.uci.edu/ml/datasets/liver+disorders, and SPECT heart dataset https://archive.ics.uci.edu/ml/datasets/spect+heart. These cases were replicated in the testing environment using the same setup as in [35] for the reasons of establishment of firm grounds for comparison.

4.1 Datasets

The Liver disorder dataset consists of records on 345 patients. The dataset covers 7 features. The Heart dataset describes cardiac single-photon emission computed tomography (SPECT) image diagnosis. Each patient is divided into two groups: normal and abnormal. To extract characteristics that summarize the original SPECT pictures, 267 SPECT image sets (patients) were analyzed. As a consequence, each patient received 44 continuous characteristics. These characteristics were then processed further to provide 22 binary features.

4.2 Metrics

Four metrics are utilized to evaluate the efficiency of the observed classifiers. The details of the performance examination metrics are as complies with [35].

(1) Accuracy: This term refers to the classifier's overall performance and is assessed as

$$Accuracy = \Big(\frac{TP + TN}{TP + FP + TN + FN}\Big) * 100 \tag{6}$$

(2) Sensitivity: It is the ratio of the number of genuine positive cases to the overall number of people who have the disease. Precision is another name for sensitivity. The sensitivity is assessed to be

$$Sensitivity/Precision = \Big(\frac{TP}{TP + FN}\Big) * 100 \tag{7}$$

(3) Specificity: This is the ratio of the total number of patients with the disease to the number of true negative instances. Recall is another name for it. Specificity is assessed as

$$Specifity/Recall = \Big(\frac{TN}{TN + FP}\Big) * 100 \tag{8}$$

Table 1. Overall metrics over the Liver dataset

Method	Best	Worst	Mean	Median	Std	Var
MLP-ISCA	**2.33E-01**	4.17E-01	3.25E-01	3.25E-01	9.22E-02	8.51E-03
MLP-SCA	2.91E-01	3.40E-01	3.20E-01	3.25E-01	1.82E-02	3.30E-04
MLP-ABC	2.43E-01	3.30E-01	**3.01E-01**	3.16E-01	3.43E-02	1.18E-03
MLP-FA	3.11E-01	3.30E-01	3.18E-01	3.16E-01	**8.05E-03**	**6.48E-05**
MLP-BA	3.20E-01	3.40E-01	3.30E-01	3.30E-01	9.71E-03	9.43E-05
MLP-HHO	2.91E-01	3.69E-01	3.35E-01	3.40E-01	2.79E-02	7.78E-04

(4) AUC: It provides a visual comparison of the true and false positive rates. The best AUC value is one with a higher value.

where TP and TN stand for the healthcare model's true positive and true negative predictions, respectively. The healthcare model's false positive and false negative predictions are denoted by FP and FN, respectively.

4.3 Experimental Results and Comparative Analysis

The proposed ISCA metaheuristics was utilized to train the MLP. Weight and bias values boundaries were set to the interval $[-1, 1]$, while the number of neurons (nn) was determined with $nn = fs * 2$ (fs denotes number of features). All features were taken into the consideration, as feature selection was not executed. The approach was simply named MLP-ISCA. The obtained results were compared to five other powerful metaheuristics, namely basic SCA ([38]), ABC ([34]), FA ([53]), BA ([54]) and HHO ([28]), that were implemented by utilizing the control parameter's setup as provided in their respective publications, and put into the test with the same goal to train the MLP.

All metaheuristics were employed with 16 solutions in the population, with 50 iterations in 15 independent runs. Each solution encoded length is given with $fs * (3 * fs) + 3 * fs$, where fs represents the features count.

The experimental results are presented in the following Tables. Table 1 shows the overall metrics of all considered approaches on the Liver dataset. As it can be noted from the table, the suggested MLP-ISCA obtained the best result for the error rate, the MLP-ABC obtained second, while the MLP-SCA and MLP-HHO were tied on third place. Table 2 presents the detailed metrics of the best run of each algorithm, and it shows the superiority of the MLP-ISCA that achieved around 76.7% accuracy on the Liver dataset, in front of MLP-ABC (75.7%) and MLP-SCA and MLP-HHO (tied on 70.9%).

Table 3 shows the overall metrics of all considered approaches on the SPECT heart dataset. As it can be noted from the table, the suggested MLP-ISCA was dominant by obtaining the best scores for all metrics concerning the error

Table 2. Detailed metrics of the best runs over the Liver dataset

	MLP-ISCA	MLP-SCA	MLP-ABC	MLP-FA	MLP-BA	MLP-HHO
Accuracy (%)	**76.699**	70.8738	75.7282	68.932	67.9612	70.8738
Precision 0	**0.827586**	0.782609	0.764706	0.593220	0.647059	0.675676
Precision 1	0.743243	0.687500	0.753623	**0.818182**	0.695652	0.727273
M.Avg. Precision	**0.778454**	0.727206	0.758250	0.724266	0.675366	0.705732
Recall 0	0.558140	0.418605	0.604651	**0.813953**	0.511628	0.581395
Recall 1	0.916667	0.916667	0.866667	0.600000	0.800000	0.800000
M.Avg. Recall	**0.766990**	0.708738	0.757282	0.689320	0.679612	0.708738
F1-score 0	0.666667	0.545455	0.675325	**0.686275**	0.571429	0.625000
F1-score 1	**0.820896**	0.785714	0.806202	0.692308	0.744186	0.761905
M.Avg. F1-score	**0.756509**	0.685412	0.751564	0.689789	0.672064	0.704750

Table 3. Overall metrics over the SPECT heart dataset

Method	Best	Worst	Mean	Median	Std	Var
MLP-ISCA	**1.20E-01**	**1.30E-01**	**1.25E-01**	**1.25E-01**	**3.63E-03**	**1.32E-05**
MLP-SCA	1.59E-01	1.98E-01	1.80E-01	1.82E-01	1.39E-02	1.92E-04
MLP-ABC	1.66E-01	2.08E-01	1.88E-01	1.88E-01	1.88E-02	3.55E-04
MLP-FA	1.49E-01	1.95E-01	1.77E-01	1.82E-01	1.68E-02	2.82E-04
MLP-BA	1.59E-01	2.08E-01	1.87E-01	1.90E-01	2.16E-02	4.66E-04
MLP-HHO	1.82E-01	2.14E-01	1.98E-01	1.98E-01	1.17E-02	1.37E-04

Table 4. Detailed metrics of the best runs over the SPECT heart dataset

	MLP-ISCA	MLP-SCA	MLP-ABC	MLP-FA	MLP-BA	MLP-HHO
Accuracy (%)	**87.9870**	84.0909	83.4416	85.0649	84.0909	81.8182
Precision 0	0.906475	0.843537	0.832215	0.876812	**0.924370**	0.797468
Precision 1	**0.857988**	0.838509	0.836478	0.829412	0.788360	0.840000
M.Avg. Precision	**0.881602**	0.840958	0.834402	0.852496	0.854598	0.819287
Recall 0	0.840000	0.826667	0.826667	0.806667	0.733333	0.840000
Recall 1	0.917722	0.854430	0.841772	0.892405	**0.943038**	0.797468
M.Avg. Recall	**0.879870**	0.840909	0.834416	0.850649	0.840909	0.818182
F1-score 0	**0.871972**	0.835017	0.829431	0.840278	0.817844	0.818182
F1-score 1	**0.886850**	0.846395	0.839117	0.859756	0.858790	0.818182
M.Avg. F1-score	**0.879604**	0.840854	0.834400	0.850270	0.838849	0.818182

rate, the MLP-FA obtained second, while the MLP-SCA and MLP-BA were tied on third place. Table 4 presents the detailed metrics of the best run of each algorithm, and it shows the dominance of the MLP-ISCA that achieved around 88% accuracy on the SPECT heart dataset, in front of MLP-FA (85.1%) and MLP-SCA and MLP-BA (tied on 84.1%). It can also be noted that the MLP-ISCA accuracy was around 3% higher than the second best approach in this particular case.

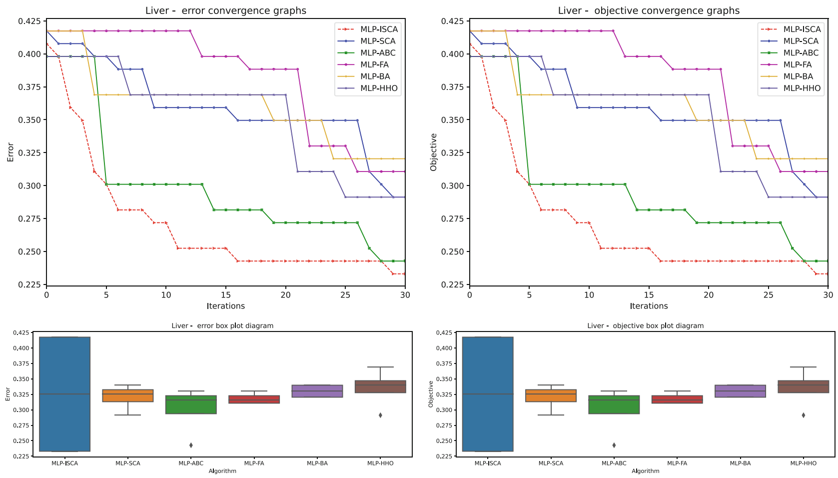

Fig. 1. Convergence and box plot graphics for all contenders on Liver dataset

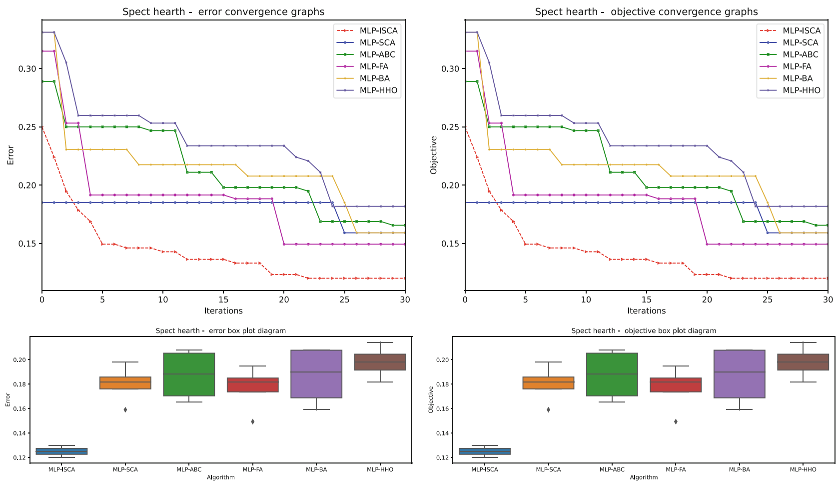

Fig. 2. Convergence and box plot graphics for all contenders on SPECT heart dataset

Graphical visualization of the superior performance of the MLP-ISCA method over other algorithms is shown in Fig. 1 and Fig. 2, that give the convergence graphs and box plots of both error rate and the objective function, of all observed methods on Liver and SPECT heart datasets, respectively. To give more detailed insight into the performance level of MLP-ISCA, Figs. 3 and 4 show precision-recall (ROC) and receiver under operating characteristics (ROC), area below the curve (AUC), confusion matrix over the Liver dataset, and one vs rest (OvR) ROC curves in comparison to other methods.

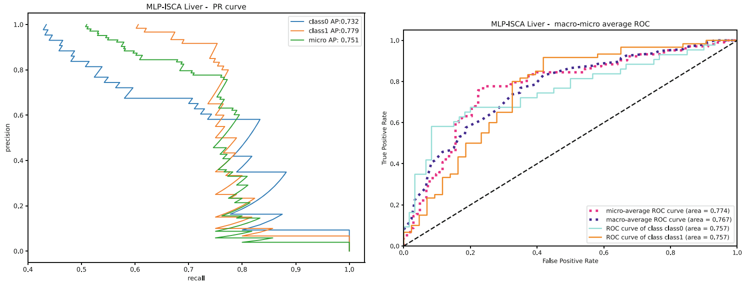

Fig. 3. PR and ROC plots of the presented MLP-ISCA model over the Liver dataset

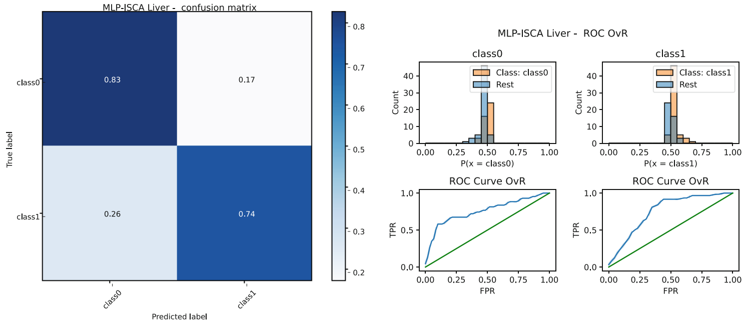

Fig. 4. The MLP-ISCA confusion matrix and one versus rest (OvR) ROC for Liver dataset

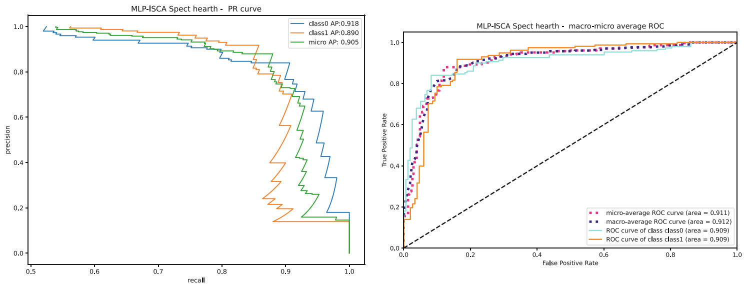

Fig. 5. PR and ROC plots of the presented MLP-ISCA model over the Heart dataset

Figures 5 and 6 show precision-recall (ROC) and receiver under operating characteristics (ROC), area below the curve (AUC), confusion matrix over the SPECT heart dataset, and one vs rest (OvR) ROC curves in comparison to other methods.

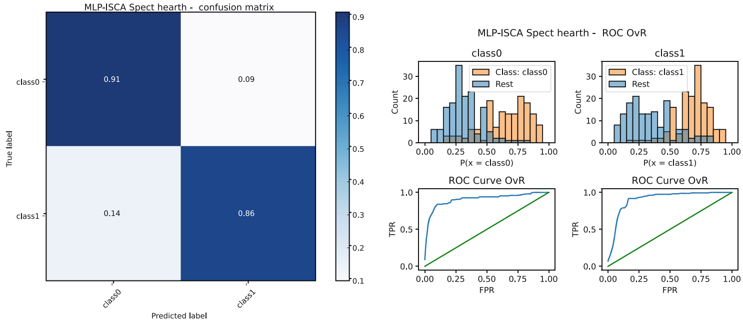

Fig. 6. The MLP-ISCA confusion matrix and one versus rest (OvR) ROC for SPECT heart dataset

5 Conclusion

This research introduced a hybrid machine learning and metaheuristics classifier with a goal to apply it to assist medical workers in disease classification. The discussed solution has been validated on two healthcare benchmark datasets, and the obtained findings have been collated to the results of the basic SCA, ABC, FA, BA and HHO algorithms that were utilized in the same setup. The experimental outcomes indicate that the suggested MLP-ISCA method achieved the highest accuracy among all competitor classifiers on both observed datasets, and show a significant potential of the approach in medical domain. The future work will focus on thorough experiments with other healthcare datasets, in order to build additional confidence in the method before applying it to the real world problems.

References

1. AlHosni, N., Jovanovic, L., Antonijevic, M., Bukumira, M., Zivkovic, M., Strumberger, I., Mani, J.P., Bacanin, N.: The xgboost model for network intrusion detection boosted by enhanced sine cosine algorithm. In: International Conference on Image Processing and Capsule Networks. pp. 213–228. Springer (2022)
2. Antonijevic, M., Strumberger, I., Lazarevic, S., Bacanin, N., Mladenovic, D., Jovanovic, D.: Robust encrypted face recognition robot based on bit slicing and fourier transform for cloud environments. Journal of Electronic Imaging 31(6), 061808 (2022)
3. Bacanin, N., Alhazmi, K., Zivkovic, M., Venkatachalam, K., Bezdan, T., Nebhen, J.: Training multi-layer perceptron with enhanced brain storm optimization metaheuristics. Comput. Mater. Contin 70, 4199–4215 (2022)
4. Bacanin, N., Antonijevic, M., Bezdan, T., Zivkovic, M., Rashid, T.A.: Wireless sensor networks localization by improved whale optimization algorithm. In: Proceedings of 2nd International Conference on Artificial Intelligence: Advances and Applications. pp. 769–783. Springer (2022)

5. Bacanin, N., Antonijevic, M., Bezdan, T., Zivkovic, M., Venkatachalam, K., Malebary, S.: Energy efficient offloading mechanism using particle swarm optimization in 5g enabled edge nodes. Cluster Computing, pp. 1–12 (2022)
6. Bacanin, N., Bezdan, T., Venkatachalam, K., Zivkovic, M., Strumberger, I., Abouhawwash, M., Ahmed, A.B.: Artificial neural networks hidden unit and weight connection optimization by quasi-refection-based learning artificial bee colony algorithm. IEEE Access 9, 169135–169155 (2021)
7. Bacanin, N., Bezdan, T., Zivkovic, M., Chhabra, A.: Weight optimization in artificial neural network training by improved monarch butterfly algorithm. In: Mobile Computing and Sustainable Informatics, pp. 397–409. Springer (2022)
8. Bacanin, N., Petrovic, A., Zivkovic, M., Bezdan, T., Chhabra, A.: Enhanced salp swarm algorithm for feature selection. In: International Conference on Intelligent and Fuzzy Systems. pp. 483–491. Springer (2021)
9. Bacanin, N., Sarac, M., Budimirovic, N., Zivkovic, M., AlZubi, A.A., Bashir, A.K.: Smart wireless health care system using graph lstm pollution prediction and dragonfly node localization. Sustainable Computing: Informatics and Systems 35, 100711 (2022)
10. Bacanin, N., Stoean, C., Zivkovic, M., Jovanovic, D., Antonijevic, M., Mladenovic, D.: Multi-swarm algorithm for extreme learning machine optimization. Sensors 22(11), 4204 (2022)
11. Bacanin, N., Stoean, R., Zivkovic, M., Petrovic, A., Rashid, T.A., Bezdan, T.: Performance of a novel chaotic firefly algorithm with enhanced exploration for tackling global optimization problems: Application for dropout regularization. Mathematics 9(21), 2705 (2021)
12. Bacanin, N., Tuba, E., Zivkovic, M., Strumberger, I., Tuba, M.: Whale optimization algorithm with exploratory move for wireless sensor networks localization. In: International conference on hybrid intelligent systems. pp. 328–338. Springer (2019)
13. Bacanin, N., Vukobrat, N., Zivkovic, M., Bezdan, T., Strumberger, I.: Improved harris hawks optimization adapted for artificial neural network training. In: International Conference on Intelligent and Fuzzy Systems. pp. 281–289. Springer (2021)
14. Bacanin, N., Zivkovic, M., Al-Turjman, F., Venkatachalam, K., Trojovskỳ, P., Strumberger, I., Bezdan, T.: Hybridized sine cosine algorithm with convolutional neural networks dropout regularization application. Scientific Reports 12(1), 1–20 (2022)
15. Bacanin, N., Zivkovic, M., Bezdan, T., Venkatachalam, K., Abouhawwash, M.: Modified firefly algorithm for workflow scheduling in cloud-edge environment. Neural Computing and Applications 34(11), 9043–9068 (2022)
16. Bacanin, N., Zivkovic, M., Jovanovic, L., Ivanovic, M., Rashid, T.A.: Training a multilayer perception for modeling stock price index predictions using modified whale optimization algorithm. In: Computational Vision and Bio-Inspired Computing, pp. 415–430. Springer (2022)
17. Bacanin, N., Zivkovic, M., Sarac, M., Petrovic, A., Strumberger, I., Antonijevic, M., Petrovic, A., Venkatachalam, K.: A novel multiswarm firefly algorithm: An application for plant classification. In: International Conference on Intelligent and Fuzzy Systems. pp. 1007–1016. Springer (2022)
18. Basha, J., Bacanin, N., Vukobrat, N., Zivkovic, M., Venkatachalam, K., Hubálovskỳ, S., Trojovskỳ, P.: Chaotic harris hawks optimization with quasi-reflection-based learning: an application to enhance cnn design. Sensors 21(19), 6654 (2021)

19. Bezdan, T., Cvetnic, D., Gajic, L., Zivkovic, M., Strumberger, I., Bacanin, N.: Feature selection by firefly algorithm with improved initialization strategy. In: 7th Conference on the Engineering of Computer Based Systems. pp. 1–8 (2021)
20. Bezdan, T., Milosevic, S., Venkatachalam, K., Zivkovic, M., Bacanin, N., Strumberger, I.: Optimizing convolutional neural network by hybridized elephant herding optimization algorithm for magnetic resonance image classification of glioma brain tumor grade. In: 2021 Zooming Innovation in Consumer Technologies Conference (ZINC). pp. 171–176. IEEE (2021)
21. Bezdan, T., Stoean, C., Naamany, A.A., Bacanin, N., Rashid, T.A., Zivkovic, M., Venkatachalam, K.: Hybrid fruit-fly optimization algorithm with k-means for text document clustering. Mathematics 9(16), 1929 (2021)
22. Bezdan, T., Zivkovic, M., Bacanin, N., Chhabra, A., Suresh, M.: Feature selection by hybrid brain storm optimization algorithm for covid-19 classification. Journal of Computational Biology (2022)
23. Bezdan, T., Zivkovic, M., Bacanin, N., Strumberger, I., Tuba, E., Tuba, M.: Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm. Journal of Intelligent & Fuzzy Systems 42(1), 411–423 (2022)
24. Bezdan, T., Zivkovic, M., Tuba, E., Strumberger, I., Bacanin, N., Tuba, M.: Glioma brain tumor grade classification from mri using convolutional neural networks designed by modified fa. In: International conference on intelligent and fuzzy systems. pp. 955–963. Springer (2020)
25. Budimirovic, N., Prabhu, E., Antonijevic, M., Zivkovic, M., Bacanin, N., Strumberger, I., Venkatachalam, K.: Covid-19 severity prediction using enhanced whale with salp swarm feature classification. Computers, Materials and Continua pp. 1685–1698 (2022)
26. Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S.: Correction: Corrigendum: Dermatologist-level classification of skin cancer with deep neural networks. Nature 546(7660), 686 (2017)
27. Hathaliya, J.J., Tanwar, S., Tyagi, S., Kumar, N.: Securing electronics healthcare records in healthcare 4.0 : A biometric-based approach. Computers and Electrical Engineering 76, 398–410 (2019), https://www.sciencedirect.com/science/article/pii/S004579061930062X
28. Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H.: Harris hawks optimization: Algorithm and applications. Future generation computer systems 97, 849–872 (2019)
29. Hu, Y.C.: Nonadditive similarity-based single-layer perceptron for multi-criteria collaborative filtering. Neurocomputing 129, 306–314 (2014)
30. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015)
31. Jovanovic, D., Antonijevic, M., Stankovic, M., Zivkovic, M., Tanaskovic, M., Bacanin, N.: Tuning machine learning models using a group search firefly algorithm for credit card fraud detection. Mathematics 10(13), 2272 (2022)
32. Jovanovic, D., Marjanovic, M., Antonijevic, M., Zivkovic, M., Budimirovic, N., Bacanin, N.: Feature selection by improved sand cat swarm optimizer for intrusion detection. In: 2022 International Conference on Artificial Intelligence in Everything (AIE). pp. 685–690. IEEE (2022)
33. Jovanovic, L., Zivkovic, M., Antonijevic, M., Jovanovic, D., Ivanovic, M., Jassim, H.S.: An emperor penguin optimizer application for medical diagnostics. In: 2022 IEEE Zooming Innovation in Consumer Technologies Conference (ZINC). pp. 191–196. IEEE (2022)

34. Karaboga, D.: Artificial bee colony algorithm. Scholarpedia 5(3), 6915 (2010)
35. Kishor, A., Chakraborty, C.: Artificial intelligence and internet of things based healthcare 4.0 monitoring system. Wireless personal communications, pp. 1–17 (2021)
36. Kumar, A., Krishnamurthi, R., Nayyar, A., Sharma, K., Grover, V., Hossain, E.: A novel smart healthcare design, simulation, and implementation using healthcare 4.0 processes. IEEE Access 8, 118433–118471 (2020)
37. Latha, R., Saravana Balaji, B., Bacanin, N., Strumberger, I., Zivkovic, M., Kabiljo, M.: Feature selection using grey wolf optimization with random differential grouping. Comput. Syst. Sci. Eng. 43(1), 317–332 (2022)
38. Mirjalili, S.: Sca: a sine cosine algorithm for solving optimization problems. Knowledge-based systems 96, 120–133 (2016)
39. Ng, A.Y.: Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In: Proceedings of the twenty-first international conference on Machine learning. p. 78 (2004)
40. Ojha, V.K., Abraham, A., Snášel, V.: Metaheuristic design of feedforward neural networks: A review of two decades of research. Engineering Applications of Artificial Intelligence 60, 97–116 (2017)
41. Petrovic, A., Bacanin, N., Zivkovic, M., Marjanovic, M., Antonijevic, M., Strumberger, I.: The adaboost approach tuned by firefly metaheuristics for fraud detection. In: 2022 IEEE World Conference on Applied Intelligence and Computing (AIC). pp. 834–839. IEEE (2022)
42. Petrovic, A., Strumberger, I., Bezdan, T., Jassim, H.S., Nassor, S.S.: Cryptocurrency price prediction by using hybrid machine learning and beetle antennae search approach. In: 2021 29th Telecommunications Forum (TELFOR). pp. 1–4. IEEE (2021)
43. Prakash, S., Kumar, M.V., Ram, S.R., Zivkovic, M., Bacanin, N., Antonijevic, M.: Hybrid glfil enhancement and encoder animal migration classification for breast cancer detection. Comput. Syst. Sci. Eng. 41(2), 735–749 (2022)
44. Salb, M., Bacanin, N., Zivkovic, M., Antonijevic, M., Marjanovic, M., Strumberger, I.: Extreme learning machine tuning by original sine cosine algorithm. In: 2022 IEEE World Conference on Applied Intelligence and Computing (AIC). pp. 143–148. IEEE (2022)
45. Salb, M., Jovanovic, L., Zivkovic, M., Tuba, E., Elsadai, A., Bacanin, N.: Training logistic regression model by enhanced moth flame optimizer for spam email classification. In: Computer Networks and Inventive Communication Technologies, pp. 753–768. Springer (2023)
46. Sreekanth, G., Ahmed, S.A.N., Sarac, M., Strumberger, I., Bacanin, N., Zivkovic, M.: Mobile fog computing by using sdn/nfv on 5g edge nodes. Comput. Syst. Sci. Eng. 41(2), 751–765 (2022)
47. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 15(1), 1929–1958 (2014)
48. Strumberger, I., Bezdan, T., Ivanovic, M., Jovanovic, L.: Improving energy usage in wireless sensor networks by whale optimization algorithm. In: 2021 29th Telecommunications Forum (TELFOR). pp. 1–4. IEEE (2021)
49. Strumberger, I., Rakic, A., Stanojlovic, S., Arandjelovic, J., Bezdan, T., Zivkovic, M., Bacanin, N.: Feature selection by hybrid binary ant lion optimizer with covid-19 dataset. In: 2021 29th Telecommunications Forum (TELFOR). pp. 1–4. IEEE (2021)

50. Strumberger, I., Tuba, E., Bacanin, N., Zivkovic, M., Beko, M., Tuba, M.: Designing convolutional neural network architecture by the firefly algorithm. In: 2019 International Young Engineers Forum (YEF-ECE). pp. 59–65. IEEE (2019)
51. Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., Fergus, R.: Regularization of neural networks using dropconnect. In: International conference on machine learning. pp. 1058–1066. PMLR (2013)
52. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE transactions on evolutionary computation 1(1), 67–82 (1997)
53. Yang, X.S.: Firefly algorithms for multimodal optimization. In: International symposium on stochastic algorithms. pp. 169–178. Springer (2009)
54. Yang, X.S., Gandomi, A.H.: Bat algorithm: a novel approach for global engineering optimization. Engineering computations (2012)
55. Zivkovic, M., Bacanin, N., Arandjelovic, J., Rakic, A., Strumberger, I., Venkatachalam, K., Joseph, P.M.: Novel harris hawks optimization and deep neural network approach for intrusion detection. In: Proceedings of International Joint Conference on Advances in Computational Intelligence. pp. 239–250. Springer (2022)
56. Zivkovic, M., Bacanin, N., Arandjelovic, J., Strumberger, I., Venkatachalam, K.: Firefly algorithm and deep neural network approach for intrusion detection. In: Applications of Artificial Intelligence and Machine Learning, pp. 1–12. Springer (2022)
57. Zivkovic, M., Bacanin, N., Tuba, E., Strumberger, I., Bezdan, T., Tuba, M.: Wireless sensor networks life time optimization based on the improved firefly algorithm. In: 2020 International Wireless Communications and Mobile Computing (IWCMC). pp. 1176–1181. IEEE (2020)
58. Zivkovic, M., Bacanin, N., Venkatachalam, K., Nayyar, A., Djordjevic, A., Strumberger, I., Al-Turjman, F.: Covid-19 cases prediction by using hybrid machine learning and beetle antennae search approach. Sustainable Cities and Society 66, 102669 (2021)
59. Zivkovic, M., Bezdan, T., Strumberger, I., Bacanin, N., Venkatachalam, K.: Improved harris hawks optimization algorithm for workflow scheduling challenge in cloud–edge environment. In: Computer Networks, Big Data and IoT, pp. 87–102. Springer (2021)
60. Zivkovic, M., Jovanovic, L., Ivanovic, M., Bacanin, N., Strumberger, I., Joseph, P.M.: Xgboost hyperparameters tuning by fitness-dependent optimizer for network intrusion detection. In: Communication and Intelligent Systems, pp. 947–962. Springer (2022)
61. Zivkovic, M., Jovanovic, L., Ivanovic, M., Krdzic, A., Bacanin, N., Strumberger, I.: Feature selection using modified sine cosine algorithm with covid-19 dataset. In: Evolutionary Computing and Mobile Sustainable Networks, pp. 15–31. Springer (2022)
62. Zivkovic, M., Petrovic, A., Bacanin, N., Milosevic, S., Veljic, V., Vesic, A.: The covid-19 images classification by mobilenetv3 and enhanced sine cosine metaheuristics. In: Mobile Computing and Sustainable Informatics, pp. 937–950. Springer (2022)
63. Zivkovic, M., Petrovic, A., Venkatachalam, K., Strumberger, I., Jassim, H.S., Bacanin, N.: Novel chaotic best firefly algorithm: Covid-19 fake news detection application. In: Advances in Swarm Intelligence, pp. 285–305. Springer (2023)
64. Zivkovic, M., Stoean, C., Petrovic, A., Bacanin, N., Strumberger, I., Zivkovic, T.: A novel method for covid-19 pandemic information fake news detection based on the arithmetic optimization algorithm. In: 2021 23rd International Symposium on

Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). pp. 259–266. IEEE (2021)
65. Zivkovic, M., Tair, M., Venkatachalam, K., Bacanin, N., Hubálovskỳ, Š., Trojovskỳ, P.: Novel hybrid firefly algorithm: an application to enhance xgboost tuning for intrusion detection classification. PeerJ Computer Science 8, e956 (2022)
66. Zivkovic, M., Zivkovic, T., Venkatachalam, K., Bacanin, N.: Enhanced dragonfly algorithm adapted for wireless sensor network lifetime optimization. In: Data intelligence and cognitive informatics, pp. 803–817. Springer (2021)